A Modular, Reconfigurable and Updateable Embedded Cyber Security Hardware Solution for Automotive

Francesc Fons, Mariano Fons, Paul Olivier, André Weimerskirch
E-SYSTEMS
Lear Corporation
Valls, Spain
{ffons, mfons, polivier, aweimerskirch}@lear.com

***Abstract*— Nowadays, the automotive industry is living a transcendental era of innovation: connected, autonomous and electric vehicles are three technology-driven megatrends that –together and at the same time– contribute to a disruptive change. Hence, great transformations are at present being addressed by the main players of automotive electronics. Car makers and suppliers try to self-adapt to this fast and radical paradigm shift by changing their mindset and, more than ever before, bringing added value in the way of new experiences to the end users, by embedding more and more data and power management inside the electronic computation units they manufacture. Nevertheless, the success of at least two of these disruptive trends –i.e., connectivity and autonomous driving– strongly depends on a critical blocker: cyber security. An enormous effort is currently addressed by the automotive stakeholders to adopt security by applying know-how from the IT industry and developing new solutions, and this happens just in a moment when traditional security is in risk due to the announced emergence, in the not-so-distant future, of quantum computers, which means that post-quantum cryptography must already be taken into consideration today when designing solutions expected to be secure for the vehicles of tomorrow. The objective of this work is to explore the best suited product architecture and technology for electronic computation units in order to handle in an efficient way –i.e., balancing security-by-design, performance and cost– the cyber security issue recently irrupted in the automotive domain as a result of the arrival of in-vehicle connectivity. Reconfigurable hardware technology combined with static multicore processors and memory, all seamlessly integrated in a single SoC device, is proposed by the authors as a strong and flexible solution to the formulated challenge, bringing moreover a clear differentiation with regard to state-of-the-art security solutions in place today in embedded cryptosystems. This work shows a real implementation example that proves the concept and provides the experimental results achieved.**

Keywords—Automotive cyber security; reconfigurable hardware; post-quantum cryptography; crypto IP core; SoC; FPGA

I. Introduction

Today’s economies are dramatically changing, triggered by the accelerated rise of new technologies. Digitization and new business models have revolutionized many industries, and automotive is not an exception. In the automotive arena, these forces are giving rise to three disruptive megatrends: autonomous driving, electrification, and connectivity [1].

At present, many engineering research initiatives related to the design and development of the next generation of embedded electronic computation units can be identified in the automotive industry which all converge in search of flexible and scalable system architectures (e.g. AUTOSAR adaptive platform, hypervisor virtualization, embedded Linux adoption, system-on-chip integration) with increasing computational power (e.g. multiprocessing, embedded computation units, cloud computing) and enhanced communication throughput (e.g. OPEN alliance, Ethernet-based network backbone, wireless connectivity, Gigabit Ethernet, CAN-FD), securely and safely deployed (e.g. SAE J3061 and ISO 26262 standards, embedded firewall, computation redundancy, cryptographic hardware accelerators like HSM, SHE and TPM cores) aimed at supporting, on top, new applications (e.g. V2X, over-the-air SW update, etc.) directly linked to those three automotive megatrends. All these engineering initiatives, once they are put together, contribute to reshape the automotive electronics industry by converging, in the end, in the deployment of what we could name an “agile” electronic computation unit, developed by multidisciplinary teams of experts through flexible engineering methodologies and tools, resulting in shortened development cycles and faster time to market. The whole automotive community is nowadays subject to all these changes and adapting to them by trying to give a fast response. One of the main drivers of such changes is the security-by-design paradigm.

The American NSA recently announced a shift from Elliptic Curve Cryptography (ECC) to post-quantum cryptography (PQC), and NIST estimates that quantum computers might be able to compromise today's public key cryptographic algorithms (e.g. RSA and ECC) or significantly reduce the security levels of symmetric key cryptographic algorithms (e.g. AES) in around 15 years with significant but feasible (for a nation attacker) financial resources. This will happen just in a moment when the cloud computing infrastructures in general (i.e. IaaS, PaaS and SaaS models) are growing, embracing more and more services and becoming the backbone of new disruptive and innovative business models, like V2X services already designed and deployed today [2]. The automotive space, that recently entered a disruptive time, requires stable and sustainable long-term cyber security solutions to eventually enable connected and automated vehicles. Not only quantum computers endanger today's automotive security solutions, but so do also advanced research results in security attacks, innovative attack tools, and flawed existing solutions.

Being aware of the fact that the efforts to replace vulnerable security solutions need to begin several years before quantum computers and other threats land, it is time to seriously motivate the use of highly flexible remotely reconfigurable and modular solutions in embedded applications, and particularly in the automotive field. For instance, most of the vehicles which are being designed today will still be on the road in more than 20 years and at that time it is assumed that quantum computers and highly advanced attack tools will be available.

In this timed race for finding a technology that solves the modeling or equation of the problem, the authors argue that the implementation of a highly flexible, modular and easily updateable security solution on programmable logic is the right choice, especially through state-of-the-art system-on-chip (SoC) devices which combine ARM core processors, heterogeneous hardware resources integrated in an FPGA and a big amount of memory, all compacted in a single chip. The change from today's cryptography to quantum resilient cryptography in deployed devices and vehicles is probably the largest and least understood challenge today. In this work, the authors describe a solution that approaches this challenge, to update today's cryptographic algorithms in deployed vehicles, and by doing so it is achieved a solution that covers a wide-majority of requirements in this space. The rationale is clear: acceleration, flexibility, personalization, security, privacy, redundancy, scalability, modularity, root-of-trust, PUF-based keys, longer key sizes, etc. are all demanded features that are inherently present in the DNA of reconfigurable hardware technology already available in SoC/FPGA devices today.

II. State-of-the-art Automotive Security Systems: Hardware Security Module

Increasing the processing power of a computation platform is a trend continuously observed not only in automotive but basically in any technological field in charge of developing embedded electronic systems (e.g. smartphones, smartcards, etc.). Security processing at run-time is one more of the contributors to this demand. As a consequence, the approach of displacing the synthesis of compute-intensive cryptographic algorithms from software to dedicated hardware is a technical decision encouraged by initiatives like EVITA (E-safety vehicle intrusion protected applications), a research project co-funded by the European Union within the Seventh Framework Programme where it is pioneered the design of the so-called Hardware Security Module (HSM) as a standard peripheral or coprocessor to be attached to automotive qualified microcontrollers so that security-relevant components are protected against tampering and sensitive data are protected against compromise [3]. Under this scope, the HSM is nowadays a dedicated customer-programmable security subsystem included in the architecture of modern automotive MCUs intended to provide advanced security features, connectable to a host processor through a communication interface so that freeing it from the execution of those compute-intensive security tasks. The HSM typically includes a secure CPU inside, along with security-specific peripherals, an AES cryptographic engine and local memories, apart from dedicated blocks of RAM used for secure code and data storage. Nevertheless, it is not difficult to notice that HSM is a static solution embedded in a MCU and, as such, it cannot withstand the frenetic pace of changes that the automotive industry triggered by the accelerated rise of new technologies, so the nature of the HSM does not allow dramatic silicon updates performed in months instead of years, which is not enough. A proof of that statement is the recent discussions about post-quantum cryptography as a replacement of the traditional cryptography in use today. In this direction, the authors believe that the original HSM concept and its architecture, as other architectures in place today like TPM and TrustZone, can be ported to a more flexible scenario based on reconfigurable hardware instead of the current MCU, ASIC or ASSP devices manufactured on static hardware, what will give rise thus to a solution that does admit fast and responsive cryptosystem updates and upgrades, also in the field, taking into account that the current security architectures will probably become obsolete in the coming years.

III. Quantum-resistant Cryptosystems

The security of the public-key cryptographic schemes relies upon mathematical problems that are assumed to be hard to solve. Currently, the most popular ones are based on the discrete logarithm problem over elliptic curves and the RSA problem. Using the existing classical computers, all known probabilistic polynomial time algorithms attempting to solve these problems only succeed with very small probability.

Shor [4] showed that, using a quantum computer, there exist efficient randomized algorithms for factoring integers and finding discrete logarithms. These algorithms take a number of steps that is polynomial in the input size. Therefore, in the case that a powerful quantum computer exists, schemes based on RSA and the discrete logarithm will become insecure. Since these schemes are widely used in all kind of communications, the existence of quantum computers will require a crucial change in communication technologies. The use of quantum computers will also affect other cryptographic primitives, but its impact is less important than in public key cryptography. Grover [5] showed that it is possible to obtain a quadratic speedup on unstructured search problems using a quantum computer. This result affects many primitives as hash functions and symmetric key encryption. However, a simple way to reduce the effect of this attack and maintain the desired level of security is to increase the key lengths.

It is still not clear if there will be quantum computers powerful enough for breaking the existing public key cryptography standards. The algorithms of Shor require a polynomial number of steps, but right now we do not have enough knowledge and power to execute them. However, many experts assume that, according to the recent advances in this area and the current investment, in the following decades there will be quantum computers with the capacity of breaking the existing public key cryptosystems. In fact, in 2015, NSA announced the plans for transitioning to quantum resistant algorithms and, in 2016, opened a call for quantum-resistant public-key cryptographic standards. These cryptographic standards are required to be secure against both quantum and classical computers and must satisfy the current and future efficiency and hardware requirements. The call will be closed in 2017, and the candidates will be analyzed for up to 5 years before being standardized. By now, there is a list of candidates whose security come from different mathematical problems: the security of the so-called code-based primitives is based on the hardness of solving a decoding problem in the context of linear codes; the security of the lattice-based primitives is based on the hardness of solving the short vector problem or the close vector problem in a lattice; other candidates base their security on solving a system of multivariate polynomial equations, finding collisions or preimages in cryptographic hash functions, or finding an unknown isogeny between a pair of supersingular elliptic curves. Thus, taking into account that attackers can try to record the current encrypted communications to decrypt them later using quantum computers, if we want to keep our communications private for a reasonable period of time, we need to switch to quantum-resistant cryptographic primitives before quantum computers come to reality.

IV. Research

This research work propels the exploration of novel system architectures and technologies to be used in the new generation of secure-by-design computation and communication units distributed inside the vehicle and interconnected through heterogeneous communication buses [6]. The work focuses on the HW/SW co-design of a secure-by-design automotive computation unit which, having as target the gateway and body domains, is composed of a full post-quantum cryptosystem implemented in programmable logic and seamlessly linked to the data path of the communication networks of the platform (e.g. gateway module or body domain controller), targeting better performance and flexibility than HSM-based MCU solutions. Apart from doing research on functional concepts, the work tries to proof them by implementing a real prototype able to deploy and exploit such concepts and then get feedback in the way of experimental results that shall help the authors take technical decisions and reach conclusions.

Reconfigurable hardware technology-based applications [7] are suggested by the authors as one of the most innovative technologies to explore in the security arena and study how they can create new opportunities. Reconfigurable hardware technology is the cornerstone of this work to encourage a sensible solution to the security problem in order to build the best possible cryptosystem based on hardware/software codesign. The fact of merging the flexibility of software engineering with the processing power of parallel and custom hardware lets the authors be confident on this technical approach. Thus, this project proposes the implementation of the whole cryptosystem around the Zynq UltraScale+ MPSoC device from Xilinx Inc. – a multiprocessing system-on-chip that combines one or more multicore ARM processors with the programmable logic resources typically found in a mid-range or high-end FPGA (from distributed flip-flops and lookup tables to embedded RAM and DSP blocks), all interconnectable through the AMBA AXI4 bus.

This article presents the work-in-progress conducted by a multidisciplinary engineering team in Lear Corporation concerning the research on secure-by-design computing platforms that combine high processing power together with flexibility and scalability. As an advanced project, all these engineering efforts deepen on new concepts that simultaneously make use of many innovative technologies:

- Hardware acceleration by means of the integration of hardware-based IP cores into the MPSoC platform. Until now, SoC/MPSoC technology has been successfully (and almost exclusively) fitted in the ADAS domain of automotive electronics. Somehow, this work explores the possibilities of extending this approach to other clusters like telecommunication, gateway and body domain controllers.
- Post-quantum cryptography. To the best of the authors' knowledge, this work pioneers the deployment of PQC-based cryptosystems for automotive. As an example, it deals with the HW implementation of the McEliece algorithm in an embedded system targeting the long term security level, aimed at reaching a cost effective solution valid for the automotive market and without sacrifying security. Apart from McEliece, other crypto IP cores are implemented following the PQC recommendations, e.g. AES-256, SHA-512 and HMAC.
- Architectural study of the interconnections of the cryptosystem (i.e. cryptographic primitives synthesized in hardware) with the communication interfaces (e.g. Ethernet, CAN-FD) aimed at reaching burst data transfers at wire-speed, achieving outstanding data throughputs where the encryption/decryption is possible to be performed at run-time –i.e., in-line and on the fly– by the custom HW crypto IP cores placed in the FPGA portion of the MPSoC device.
- Flexibility concerning agile adaptation to security changes (quick response to cryptographic algorithms upgrade in the field in case the security gets compromised, with no ECU replacement required), and highest degree of personalization (customization of security features based on car configuration parameters). Moreover, such product must escalate well in order to accommodate not only high-end but also mid-range and low-end versions of the solution.

- Applications partitioning. Aimed at developing applications where it is possible to coexist safety and non-safety relevant functions, the usage of a hypervisor is a good approach to isolate for instance ASIL-D functions from other functions with lower automotive safety integrity levels like ASIL-A or also QM, as well as guaranteeing the freedom of interference among them.
- Security layers: Firewall, MACsec, IPsec layers integrated in Embedded Linux as part of one of the hypervisor partitions of the system.
- Deployment of applications like V2X (e.g. high demand of authentications per second in V2V) or Over-The-Air SW update (SOTA) which require security services.

V. DESIGN

In this section, it is provided some rationale about the main technical decisions taken at the moment of defining the product - an all-in-one secure computing unit (with a full cryptosystem included) oriented to embed functions from both gateway and body domains, and inspired by the high level of performance demanded nowadays to the ADAS modules.

A. *Quantum-resistant algorithm candidates*

The cryptographic primitives implemented through HW/SW co-design methodologies and included in the cryptosystem have been accurately selected following the recommendations of post-quantum cryptography. Such dedicated hardware security components or cryptographic accelerators encapsulate security functions aimed at providing the necessary root-of-trust to the full system.

1) *Symmetric-Key Cryptosystems*

According to the document NIST SP 800-57 “Recommendation for Key Management” [8] and [9], the cryptosystem AES-128, with keys of length 128 bits, attains 128 bits of security in the context of conventional computing and 64 bits of security in the context of quantum computing. AES-256, with keys of length 256 bits, attains 256 bits of security in the context of conventional computing and 128 bits of security in the context of quantum computing. The need of doubling the key size is a consequence of the Grover attack [5]. This attack is based on a quadratic speedup on unstructured search problems using a quantum computing. Therefore, by default, the key sizes of cryptosystems have to be doubled in order to obtain the same security in the context of quantum computing. For hash functions, this change in the key is translated into a need of larger outputs. According to [8], in order to have 256 bits of security, it is enough to use SHA-512 or SHA3-512 for hash-only applications, and SHA-256, SHA-512/256, SHA-384 or SHA-512, SHA3-512 for HMAC, Key Derivation Functions and Random Number Generation.

Following these recommendations, the AES-256 and the SHA-512 cores have been chosen to be implemented in this work in order to meet a PQC-compliant cryptosystem. Apart from that, the AES-128 and the SHA-256 have also been synthesized but only for comparison reasons, in order to benchmark our solution with state-of-the-art implementations available today in automotive MCUs from several semiconductor vendors (e.g. Renesas, Infineon, NXP) as outcome of the EVITA project.

2) *Public-Key Cryptosystems*

McEliece [10] presented in 1978 a code-based public-key cryptosystem. It has not been used in practice until now because it requires keys much larger than other cryptosystems based on RSA and ECC. The McEliece cryptosystem can attain the same level of security against attacks performed by classic computers as the ones based on RSA and elliptic curves. Moreover, the McEliece cryptosystem is also considered secure even against quantum attacks. Hence it is a clear candidate for quantum-resistant public-key cryptographic standard. Also, there are variants of the McEliece cryptosystem attaining CCA-2 security.

Apart from the symmetric and public algorithms selected, additional security-related components like an internal memory to store the keys, a TRNG core, a timer and a DMA interface to connect the raw data memories with the cryptographic and communication cores have been synthesized in the programmable logic and connected to the system bus, as detailed next in this section.

B. *Technology choice*

The target device of this design is the Xilinx XCZU9EG MPSoC, a member of the Zynq UltraScale+ family. The MPSoC is divided in two main parts: Processing System (PS) and Programmable Logic (PL), as illustrated in Figure 1.

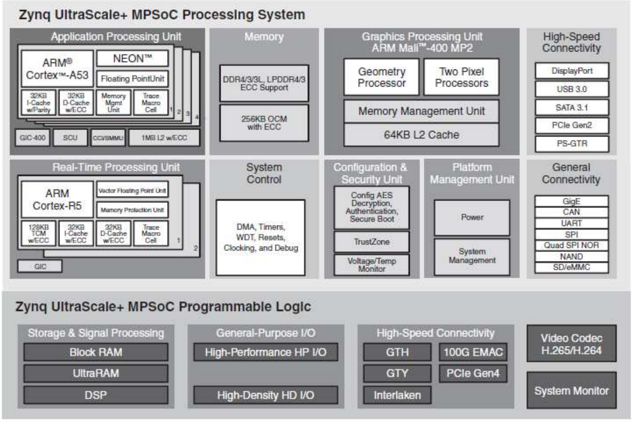

Figure 1. Zynq UltraScale+ MPSoC block diagram (Xilinx source)

Main features of the PS are the following:

- Application Processor – Quad-core ARM Cortex-A53.
- Real-time Processor – Dual-core ARM Cortex-R5.
- GPU – ARM Mali-400.
- External memory I/F – DDR4, LPDDR4, DDR3, etc.
- High-speed connectivity – Tri-mode Gigabit Ethernet, PCIe Gen2x4, USB3.0, etc.
- Security – AES, RSA and SHA cores.

Regarding PL, it can be seen as a sea of programmable resources distributed inside the FPGA area of the MPSoC that are configured by downloading a bitstream.

Apart from that, the MPSoC is equipped with a powerful set of features that clearly contribute, by design, to run-time security, as collected in Table I.

TABLE I. MAIN RUN-TIME SECURITY FEATURES OF MPSOC

MPSoC	Contribution to Run-Time Security
XMPU	Partition access to specific memory regions to specific system masters (CPU and DMA)
XPPU	Partition configuration and control of specific peripherals to specific system masters
SMMU	Constrain memory regions accessible by DMA masters
TrustZone	Partition system across secure and non-secure domains
OCM, TCM, BRAM	Hide sensitive data within internal memory
DMA	Constrain DMA accesses on a per-channel basis

C. *System Architecture*

In the eyes of the authors, reconfigurable hardware arises as the firmest technology candidate to balance in the proper weights strong design requirements like high-performance computing (DMIPS), security level and flexibility demanded to the target product. By making use of the hardware/software co-design methodology, it is possible to synthesize in hardware the most demanding (from the security and processing perspective) functionality of the algorithms embedded into a high-end automotive computation unit whereas the remainder parts are left in software. According to these criteria, Figure 2 shows the proposed system architecture.

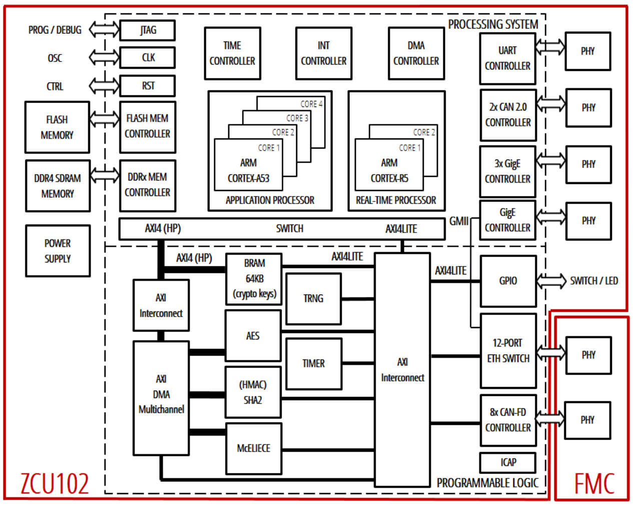

Figure 2. System block diagram of the electronic computation unit

The designed product meets the functions of a gateway and body controller from the perspective of communications and security. Figure 3 shows the physical prototype developed in this project. Thanks to this heterogeneous architecture, those compute-intensive tasks with hard real-time constraints such as response time to some events can be supported directly in hardware by specific made-to-measure peripherals synthesized in the programmable logic of the MPSoC device and then handled in software from the application that runs in the ARM multicore processor. The fact that the AXI4 bus is accessible to the programmable logic enables the option of implementing specific coprocessors that reach a seamless and effective connection to the ARM processors.

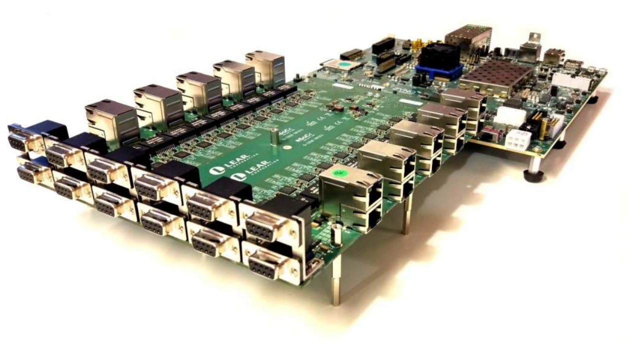

Figure 3. Lear hardware platform based on the Xilinx ZCU102 evaluation board and extended through a custom FMC-based communications board developed

VI. DEVELOPMENT

In this section it is described how the cryptosystem is architected. As a first overview, it is composed of three main parts: (i) the PS part of the MPSoC device, composed by the multicore processors with their standard peripherals attached, and a customized subsystem developed inside the PL of the MPSoC; this extended PL-based subsystem can be divided, in its turn, as a region with two different functions: (ii) cryptographic hardware accelerators and (iii) communication hardware controllers. In the end, the full system constitutes a computation unit in search for the right architecture and technology to deliver root-of-trust and performance to process specific automotive applications.

Next it is described the two flexible parts that have been developed in the PL of the MPSoC device:

- Crypto IP cores to support the high-demanding computations, aimed at performing wire speed data encryption/decryption and data authentication/signature, as illustrated in the left side of the PL in Figure 2.
- Combination of a set of heterogeneous communication controllers, to be exact, CAN-FD and Ethernet, as depicted in the right side of the PL in Figure 2.

A. *Cryptographic primitives*

All the crypto IP cores implemented in this work keep the same architecture concerning data management: they are implemented as coprocessors connectable to the AMBA AXI4 bus in order to enable on-chip communication with the ARM core technology. Internally, they all are composed of a processing core and a set of configuration and control/status registers which can be accessed by any master processor connected to the bus. Moreover, all these cores are designed to deliver a high level of flexibility through the parameterization of all their critical features identified. This agility, inherent to the reconfigurable hardware technology, is welcome in those scenarios with very changing environmental conditions like automotive today.

1) True Random Number Generator

Generating random numbers of quality is a key point in any cryptosystem. Random number generators can be assessed according to three factors: the nature or source of randomness, the latency time spent to generate a random number and the security level of its physical implementation.

Making use of the programmable logic available in the FPGA portion of our MPSoC platform, the TRNG developed in this work is based on the implementation of generalized ring oscillators (GRO) as source of randomness (i.e., an asynchronous circuit which provides a chaotic behavior based on the slight differences between the logic delays and paths through the XOR gates and the noise present in any digital circuit). The original random numbers generated by the GRO circuitry are then sourced to a linear hybrid cellular automaton (LHCA) that works as a scrambler to deliver a uniform distribution of random numbers, as proposed in [11].

In our custom implementation, the True Random Number Generator is a parameterized IP core with flexible features. Like this, its synthesis can be customized to generate random numbers of different sizes: 32, 64, 128 and 256 bits. The generation time is also a parameter that can be configured by the architect of the cryptosystem, selecting the number of clock cycles that the GRO oscillates asynchronously before the LHCA gets the produced raw data to generate the definitive random number.

As depicted in Figure 4, the IP core is connected to the AXI4 bus as a slave peripheral, being thus accessible by any master processor connected there (e.g. ARM-core processors). The block diagram of the TRNG is shown next, decomposed in the two functional blocks: GRO and LHCA.

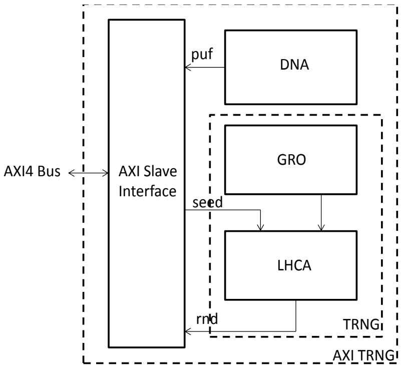

Figure 4. Block diagram of the Lear AXI4-based TRNG IP core

This TRNG core is used in the cryptosystem to generate keys in algorithms like AES or in McEliece (Goppa codes).

2) Advanced Encryption Standard

According to [8], in symmetric systems under Grover's attack, the best security a key of length n can offer is $2^{n/2}$, therefore, although AES-128 attains 128 bits of security in the context of conventional computing, it only achieves 64 bits of security in the context of quantum computing. PQCRYPTO [12] recommends thoroughly analyzed ciphers with 256-bit keys, i.e., AES-256, to achieve 2^{128} post-quantum security.

The Advanced Encryption Standard (AES) IP core developed in this work is a parameterized coprocessor synthesizable in the way that can perform at 128-bit or 256-bit. Independently of the two possible instantiations of the IP core, it can work in one of up to three cipher/decipher modes: ECB, CBC and GCM according to its configuration parameters. The IP core is delivered as a slave peripheral attachable to the AXI4-Lite bus for accessing to its configuration registers, while the data path for encryption/decryption of raw data is managed through a master and slave AXI4-Stream interface from where an external device can retrieve/store the encrypted and/or decrypted data streams from/to memory, typically making use of the AXI DMA controller. The block diagram of the flexible AES IP core is illustrated in Figure 5.

The AES IP core works on data blocks of 128 bits independently of the data size chosen in its synthesis or instantiation (128 or 256-bit) and of its working mode (ECB, CBC or GCM) selected in its configuration. Therefore, given a 128-bit input data block, the core computes a 128-bit output block according to the configured mode, taking into account the key and initial value (IV) given, after a specific number of iterations (11 rounds for 128-bit key or 15 rounds for 256-bit key). The round keys are precomputed in the Key Generator block when required.

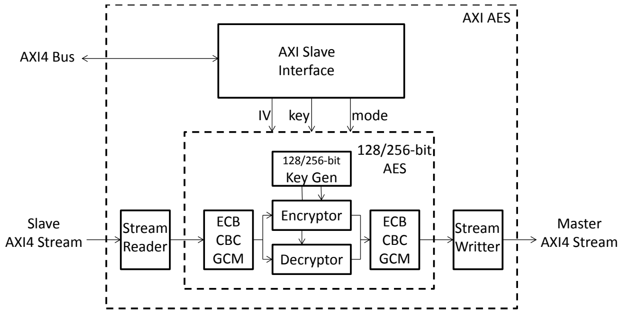

Figure 5. Block diagram of the Lear AXI4-based AES IP core

3) Secure Hash Algorithm

As highlighted in section V, according to [8], in order to have 256 bits of security, it is enough to use SHA-512 or SHA3-512 for hash-only applications, and SHA-256, SHA-512/256, SHA-384 or SHA-512, SHA3-512 for HMAC, Key Derivation Functions and Random Number Generation.

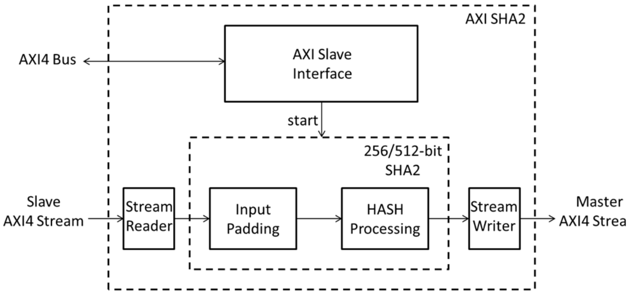

Figure 6. Block diagram of the Lear AXI4-based SHA2 IP core

In this work, from the different Secure Hash Algorithm (SHA) options mentioned above, it has been implemented the SHA-256, SHA-512 and then the HMAC-SHA2 IP cores. In this case, SHA-256 and SHA-512 are two independent cores due to the internal differences in structure that, according to the designers, would carry important constraints in both area and speed if trying to communalize in only one design. Nevertheless, the IP core structure can be clearly decomposed in two parts or components, as shown in Figure 6: the input/padding unit and the hash core. From these two parts, the reading/padding unit keeps essentially identical in both versions of the SHA2 IP core implemented and can be selected by configuration registers. Also, the external interfaces do not differ in both SHA2 IP cores. Regarding the HMAC-SHA2 IP core, it is based on the previous HW SHA2 IP core together with additional functions that are synthesized directly in SW and executed by the ARM processor.

4) McEliece

The McEliece algorithm belongs to the family of so-called code-based cryptosystems. The one-way function associated to this kind of cryptosystems is the addition of errors to codewords of a linear code. The trapdoor of this one-way function is the knowledge of an efficient error-correcting algorithm for the considered family of linear codes and the knowledge of a secret matrix permutation. The McEliece cryptosystem is considered the first code-based cryptosystem and was originally based on binary Goppa codes. The private key is a random binary irreducible Goppa code and the public key is a random generator matrix of a randomly permuted version of the code. The ciphertext is a codeword to which some errors are added, and only the owner of the private key can remove those errors efficiently.

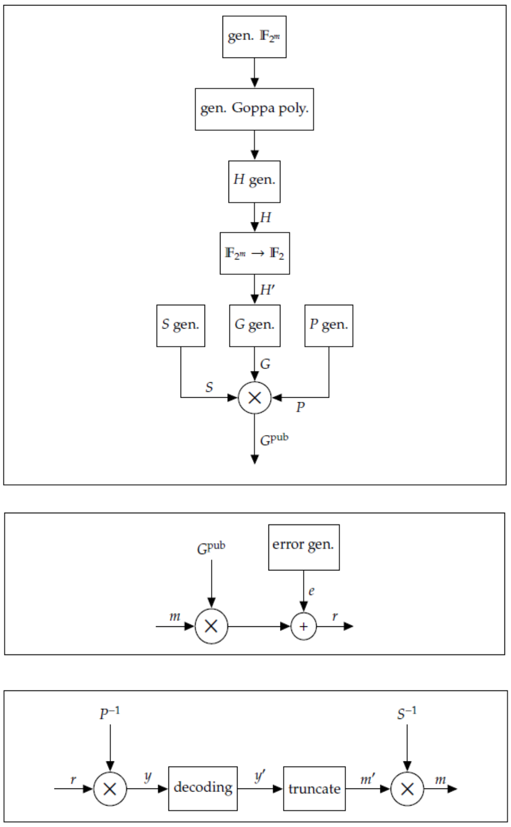

Figure 7. McEliece computation: key generation, encryption & decryption

In the context of the PQCRYPTO "Post-quantum cryptography for long-term security" project –a Horizon 2020 (H2020) project aimed at providing recommendations and software for next quantum-resistant cryptosystems– the initial recommendation [12, 13] for Public-key encryption is to use McEliece with binary Goppa codes of length 6960, dimension 5413 and error threshold 119. Although other candidates are being evaluated, currently the McEliece cryptosystem is the only one that is recommended.

To the best of the authors' knowledge, this work brings the pioneer hardware implementation of the McEliece algorithm targeting an embedded system and addressing the highest level of security (i.e., long-term security, 256 bits or more). The hardware platform gets reduced to the PS and PL parts of the MPSoC device, together with external memory (DDR-SDRAM and Flash) for data storage.

The algorithm is decomposed in three parts which are codesigned in hardware and software: Goppa codes generation, encryption and decryption [14]. According to the software profiling obtained and the timing requirements of each part, the keys generation (Goppa codes) can be kept implemented directly in software while the encryption and decryption steps are performed in hardware achieving thus an efficient algorithm execution time in line with the application requirements. Figure 7 shows the processing flow of McEliece algorithm, covering the key generation, encryption and decryption.

B. Communication Controllers

Next, it is briefly described the communication interfaces that have been synthesized in the programmable logic of the MPSoC device, aimed at providing a platform where crypto IP cores and communication IP cores are seamlessly connected to each other, together with memory data buffers for both ciphertext and plaintext storage, in order to deliver a good level of performance. The communication channels established are Ethernet and CAN-FD, as depicted next in Figure 8.

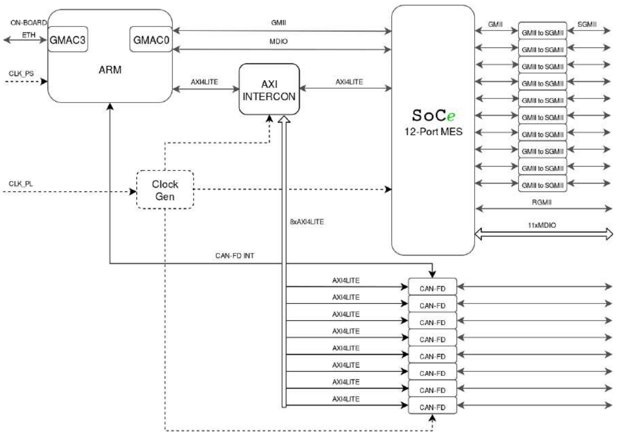

Figure 8. Block diagram of the communications subsystem

1) Gigabit Ethernet Switch

GMAC0 Gigabit Ethernet controller from the PS is internally connected to the SoC-e Managed Ethernet Switch (MES) IP core instantiated in the PL, providing thus Ethernet access to the ARM core processors. These processors also interact with the MES IP core through the AXI4-Lite bus.

SoC-e Managed Ethernet Switch provides 12 Ethernet ports as it can be seen in Figure 8. One of them is connected to GMAC0 of the PS section and the other eleven are routed to the pins linked to the programmable logic portion of the MPSoC device. Due to the used hardware, SGMII outputs are needed, so MES ports are not directly routed to pins. To adapt GMII to SGMII interface, Xilinx Ethernet SGMII modules are used: these modules instantiate available MGTs in the PL section to serialize data and convert GMII into SGMII, as detailed in Figure 8. There is also one port in RGMII mode which is directly routed to pins. Moreover, MES is also connected to PS section through the AXI4-Lite bus. This bus is used for configuration and status monitoring. A more detailed structure of the Ethernet switch is illustrated in Figure 9, where the core of the switch is a non-blocking crossbar matrix that allows continuous transfers between all the ports.

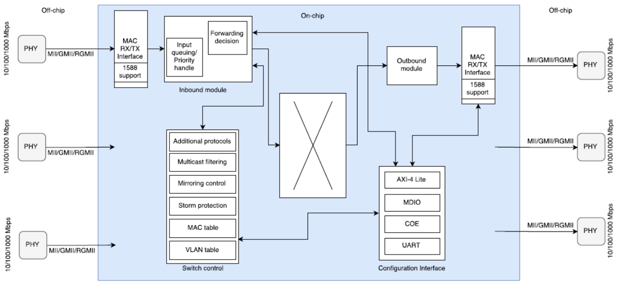

Figure 9. Ethernet switch core architecture (SoC-e source)

2) *CAN-FD*

PS section of the MPSoC device also has access to CAN-FD controllers via AXI4-Lite bus, as shown in Figure 8. These CAN-FD modules have interrupt ports connected to PS and CAN-FD ports directly routed to programmable logic-related pins.

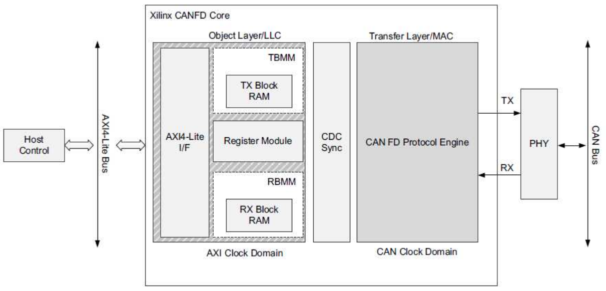

Figure 10. CAN-FD core layered architecture (Xilinx source)

Figure 10 illustrates the high-level architecture of the CAN-FD core and provides the interface connectivity. The core functions of the CAN-FD module are divided into two independent layers: (i) the object layer, which works in the AXI4 clock domain, interfaces with the host control through the AXI4-Lite bus and provides a transmission and reception method to manage message buffers; (ii) the transfer layer, which operates in the CAN clock domain, deploys the protocol engine and interfaces with the external PHY. Information exchange between the two layers is done through the CDC synchronizers.

In the authors' opinion, this product architecture delivers an outstanding level of integration (12-port Ethernet switch and 8 CAN-FD buses in a single chip) and is much more compact and scalable than other MCU-based alternative architectures based on two or more MCU devices with discrete Ethernet switch components, fact that allows to skip, by design, potential synchronization and latency issues among the system buses.

C. *Applications*

At the top of the system we can find several potential applications that incorporate security. A typical use case would be the implementation of a Connected Gateway Module which merges communication links (Ethernet, CAN, etc.) with cryptographic accelerators (AES, SHA, etc.), running applications like Over-The-Air SW Update (SOTA), Vehicle-to-Vehicle or Vehicle-to-Infrastructure (V2X), automotive Ethernet/CAN firewall, or even safety-related functions of up to ASIL-D level (e.g. steering column lock).

In the specific case of having critical safety-related functions (typically ASIL C and D) running together with lower level safety (ASIL A and B) or non-safety (QM) functions in the same platform, due to security and safety reasons, it can be convenient to establish a partitioning of applications through a virtualization layer implemented by means of a hypervisor. This scenario has been considered also in the scope of this advanced project in order to investigate its feasibility. In this direction, the Sysgo PikeOS hypervisor has been selected and integrated into the Zynq UltraScale+ MPSoC system in order to isolate two partitions, from where not only the processor peripherals of the MPSoC device but also the customized coprocessors or IP cores integrated in the programmable logic can be managed securely by both partitions through specific APIs, as shown in Figure 11.

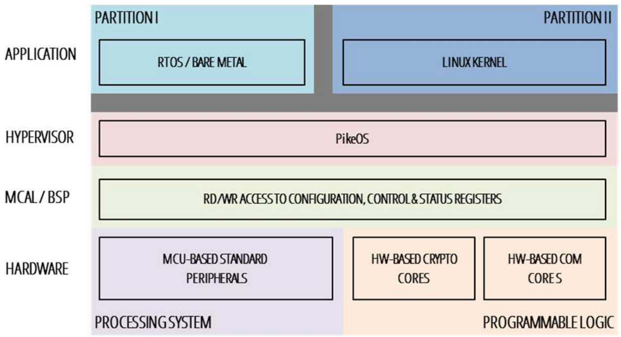

Figure 11. SW stack of the full HW/SW system

VII. EXPERIMENTAL RESULTS

This section highlights the more relevant results achieved through the implementation of the proposed PQC-based cryptosystem making use of HW/SW codesign under the physical platform developed by Lear Corp. based on the Xilinx Zynq UltraScale+ MPSoC ZCU102 evaluation board shown in Figure 3. It is important to notice that some results are still not available at the moment of writing this article, targeting, however, to be available at the time of the oral presentation of this work in the embedded world conference in Nuremberg in March 2017. The following tables show the experimental results related to resources and time obtained along all the tests performed until now. The specific MPSoC device in use is XCZU9EG-2FFVB1156I.

TABLE II. HW RESOURCES OF THE CRYPTOGRAPHIC IP CORES

HW Resources	Used	Available	Util %
CLB	2205	34260	6.44
LUT as Logic	11236	274080	4.1
LUT as Memory (Distributed RAM, Shift Register)	1054	144000	0.73
LUT Flip Flop Pairs	3399	274080	1.24
BRAM	3.5	912	0.38
DSP Block	0	2520	0

Table II summarizes the hardware resources used by the suite of cryptographic IP cores synthetized in the FPGA, namely: SHA-256/SHA-512, AES-128/AES-256 and TRNG, along with the interfaces with the DMA, timer and so on, as illustrated in the left side of the PL in Figure 2.

TABLE III. HW RESOURCES OF THE COMMUNICATION IP CORES

HW Resources	Used	Available	Util %
CLB	9847	34260	28.74
LUT as Logic	37995	274080	13.86
LUT as Memory (Distributed RAM, Shift Register)	1228	144000	0.85
LUT Flip Flop Pairs	13511	274080	4.93
BRAM	119.5	912	13.10
DSP Block	0	2520	0

Analogously, Table III summarizes the hardware resources used by the communication IP cores, i.e., a 12-Ports Ethernet Switch and 8 CAN-FD controllers and the interconnects, as depicted in Figure 8 and also in the right side of the PL in Figure 2.

TABLE IV. TIME PERFORMANCE OF THE SW-ONLY IMPLEMENTATION

Cryptographic Operation	Block Size (bits)	Throughput (MB/s)	Timing (ms)
SHA-256	512	34.5	33.50
SHA-512	1024	46.5	24.86
AES-128	128	3.81	202.24
AES-256		2.90	398.40
McEliece encryption	k=17 (n=32, t=3, m=5)	2.13	469.49
McEliece decryption		1.95E-02	51284.97
McEliece encryption	k=260 (n=512, t=28, m=9)	3.49	286.93
McEliece decryption		3.15E-04	3175425.97
McEliece encryption	k=2056 (n=4096, t=170, m=12)	4.31	231.79
McEliece decryption		6.17E-06	162172316.88

Table IV shows the execution times of the SHA2, AES and McEliece algorithms implemented only in software. The tests are performed in one of the four ARM Cortex-A53 cores available in the MPSoC device of the Xilinx ZCU102 evaluation board running at a clock frequency of 1.099989014 GHz and processing 1.2113161MB of data.

TABLE V. TIME PERFORMANCE OF THE HW/SW IMPLEMENTATION

Cryptographic Operation	Block Size (bits)	Throughput (MB/s)	Timing (ms)
SHA-256	512	272.9	4.24
SHA-512	1024	376.8	3.07
AES-128	128	415.3	2.78
AES-256		304.7	3.79

Analogously, Table V shows the execution times of SHA2 and AES implemented in hardware through crypto IP cores at a clock frequency of 300MHz when processing 1.2113161 MB of data. As observed, the acceleration ratio with respect to the SW-only implementation is around one or two orders of magnitude.

TABLE VI. HSM PERFORMANCE BENCHMARK

Crypto. Operation	Technology	Block Size (bits)	Cycles/ Block	Clock Speed (MHz)	Throughput (MB/s)
SHA-256	Infineon Aurix	512	64	100	94,00
	ST Chorus 8M		66	100	92,48
	ST Stellar		66	200	184,96
	Renesas R-Car		-	400	329,00
	MPSoC (Lear)		64	300	272.9
SHA-512	ST Chorus 8M	1024	97	100	125,85
	ST Stellar		97	200	251,69
	Renesas R-Car		-	400	469,00
	MPSoC (Lear)		82	300	376.8
AES-128	Infineon Aurix	128	14	100	108,99
	ST Chorus 8M		12	100	127,16
	ST Stellar		12	200	254,31
	Renesas R-Car		11	400	554,87
	MPSoC (Lear)		11	300	415.3
AES-256	ST Chorus 8M	128	16	100	95,37
	ST Stellar		16	200	190,73
	Renesas R-Car		15	400	406,90
	MPSoC (Lear)		15	300	304.7

Finally, as benchmark, Table VI shows the maximum throughput of some HSMs from different MCU vendors compared to our hardware-based solution implemented in the MPSoC device. As highlighted there, our developed IP cores perform, for each cryptographic computation, the same or less number of clocks than the fastest HSMs available in the industry today.

VIII. Conclusions

Processing efficiency, scalability, and flexibility of cryptosystems are more important now than ever before. The combination of software code running on the CPU cores with critical sections of compute-intensive cryptographic applications processed directly in hardware brings a technological differentiation that delivers a clear competitive advantage to cryptosystems. Furthermore, the flexibility and system integration capabilities provided by reconfigurable hardware makes it a promising technology to fulfill the stringent and changing requirements that nowadays are committed to automotive computing and security systems.

The merging of core processors with programmable logic together with large amounts of dedicated memory, all in a single chip device makes SoC technology an attractive platform to develop a secure-by-design cryptosystem-on-chip where, moreover, the HW-based crypto IP cores can be efficiently connected to the communication IP cores in order to reach wire-speed bursts of data, delivering a flexible and very agile solution ready to in-vehicle upgrades of the cryptographic algorithms without requiring any replacement of the electronic control unit (ECU) but only reprogramming it as soon as the post-quantum cryptography standards get consolidated in the coming years.

In the same way as the upgradeability performance of software is widely consolidated today in many embedded system applications, authors believe that programmable and flexible hardware will see a broader and deeper adoption in the short future in many fields of electronic embedded systems, but especially in the automotive industry, delivered in the form of highly-flexible hardware and software electronic control and computation units, mainly due to the unquestionable flexibility this technology is able to deliver.

Acknowledgment

This article takes part of a patent-pending disclosure provided by Lear Corporation as a result of the work-in-progress conducted by a development team from the Valls Engineering Center (Spain) under the format of an advanced research project executed from October 2016 to March 2017. Authors would like to thank all the multidisciplinary teams who, in one way or another, have supported and made relevant contributions to this work: Dr. Enrique Cantó and Dr. Oriol Farràs from Universitat Rovira i Virgili; Dr. Mariano López and Dr. Rubén Lumbiarres from Universitat Politècnica de Catalunya; Dr. Armando Astarloa, Mikel Idirin, Alain García, Sergio Salas, Aitor Rodríquez and Dr. Jesús Lázaro from System-on-Chip Engineering S.L. (SoC-e); Benjamin Reynaud, Eric Díaz, Bertrand Marquis and Chris Berg from Sysgo AG; Stephan Janouch, Mark Harvey and Jeff Bochenek from Xilinx Inc.; Ricardo Gómez-Galarza from Avnet Silica Inc.; and, last but not least, Juan Urbelino Nava, Oliver Ruíz, Marcial Quintanilla, Lorenzo Blasco, Alba Pruñonosa, Adrià Garcia, Elisabeth Ibarra, Ricardo Martínez and José Gabriel Fernández from Lear Corporation. Thanks all for your energy.

References

[1] McKynsey&Company, Automotive revolution – perspective towards 2030, January 2016.

[2] F. Fons, Next disruptive digital business model innovation: neuroplastic clouds, Xcell Software Journal, Issue 3, first quarter 2016, pp.40-46.

[3] EVITA. E-safety vehicle intrusion protected applications. FP7-ICT-2007-2, http://evita-project.org/.

[4] P. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput., 26 (5), 1997, pp. 1484–1509.

[5] L. Grover, A fast quantum mechanical algorithm for database search, Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC 1996), 1996, pp. 212–219.

[6] F. Fons and M. Fons, FPGA-based automotive ECU design addresses AUTOSAR and ISO 26262 standards, Xcell Journal, Issue 78, first quarter 2012, pp. 20-31.

[7] F. Fons and M. Fons, Making biometrics the killer app of FPGA dynamic partial reconfiguration, Xcell Journal, Issue 72, third quarter 2010, pp. 24-31.

[8] Recommendation for Key March, 2007 Management Part 1: General (Revised) SP 800-57 Part 1, Rev. 4 (as of January 29, 2016).

[9] Report on Post-Quantum Cryptography. National Institute of Standards and Technology Internal Report 8105, April 2016, http://dx.doi.org/10.6028/NIST.IR.8105.

[10] R. McEliece, A public key cryptosystem based on algebraic coding theory, DSN progress report 42.44, 1978.

[11] Catalin Baetoniu, High speed true random number generators in Xilinx FPGAs, Xilinx.

[12] D. Augot et al., Initial recommendations of long-term secure postquantum systems. PQCRYPTO: Post-Quantum Cryptography for Long-Term Security. Horizon 2020 ICT-645622, Revision 1, September 2015.

[13] D.J. Bernstein and T. Lange, Long-term security for cars, Invited talk at ESCAR (Embedded Security for Cars) EU 2016 conference, Munich, November 2016.

[14] D.J. Bernstein, J. Buchmann and E. Dahmen (eds.), Post-Quantum Cryptography, Springer-Verlag Berlin Heidelberg, 2009.